METHOD AND SYSTEM FOR SECURE AUTHENTICATION

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 60/459,508, filed March 31, 2003, entitled METHOD AND SYSTEM FOR PROTECTING INFORMATION ENTERED VIA INTERNET WEB PAGES, hereby incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

[0002] The present disclosure generally relates to protecting information transmitted in a computer network and, more specifically, to protecting passcode information transmitted via the Internet.

[0003] Electronic commerce, also referred to as e-commerce, provides the opportunity for merchants to reach consumers that are outside of a geographic region normally associated with a physical storefront. However, a merchant implementing electronic commerce needs to address the associated issues of customer payment, payment authentication, and payment authorization.

[0004] A merchant can accept, for example, customer credit cards or debit cards for payment. The merchant may authenticate a payment card presented by a customer in a physical storefront. The merchant can compare a customer signature in a field on the back of the card to the signature provided on a sales receipt. Additionally, the merchant may authenticate the customer by asking for an independent picture identification such as a driver's license. A merchant may also electronically authenticate a payment card received from a customer at a physical storefront. The merchant can read information typically stored on a magnetic stripe on the back of the payment card. The merchant can read the information on a point of sale device located at the checkout counter. The point of sale device can be connected to an issuer network. The payment card issuer can authenticate the payment card by comparing the information stored on the magnetic stripe to corresponding information stored in an issuer database. The point of sale device can be configured to provide an additional measure of security by asking for a Personal

Identification Number (PIN) or passcode associated with the payment card. Presumably, only an authorized user has access to the passcode. The customer maintains control over the payment card and passcode throughout the transaction and the merchant typically has no ability to access the customer's passcode. The customer provides the passcode directly to the point of sale device, which is connected to the payment card issuer network.

[0005] A merchant and customer engaged in an electronic commerce transaction complicate the authentication process. The merchant does not have physical access to the payment card. Additionally, a customer may be hesitant to supply a passcode to a merchant.

[0006] Secure communication links, such as Secure Sockets Layer (SSL) connections authenticate the parties and secure the information provided using the connection. However, such security protocols do not contribute to authenticating a payment card. Additionally, such security protocols do not provide a consumer with any level of confidence that a passcode is not stored in unencrypted form on a merchant server or database.

[0007] Hence it is desirable to provide a system for authenticating a payment card that securely maintains consumer personal information such as passcode information. The authentication system should allow secure payment card authentication and should not compromise the security of any underlying authentication networks.

BRIEF SUMMARY OF THE INVENTION

[0008] A system and method to provide secure Personal Identification Number (PIN) based authentication is disclosed. A passcode or PIN associated with a customer value card can be securely authenticated by an issuer prior to authorizing payment. An Access Control Server (ACS) can receive the PIN or passcode from a customer via a secure connection over a public network. The ACS can generate an encrypted PIN and can communicate the encrypted PIN to a remote issuer for authentication. The ACS can use one or more hardware security modules to generate the encrypted PIN. The hardware security modules can be emulated in software or implemented in hardware. The system can be configured such that the PIN is not exposed in an unencrypted form in a communication link or in hardware other than the originating customer terminal.

[0009] In one aspect, the disclosure includes a secure passcode authentication system. The system can include an Access Control Server (ACS) configured to receive a request for passcode authentication of a Primary Account Number (PAN), and configured to request a passcode corresponding to the PAN, a front end Hardware Security Module (HSM) coupled to the ACS, and configured to receive the passcode and generate an encrypted passcode using a local encryption key, and a back end HSM configured to receive the encrypted passcode from the front end HSM and further configured to recover a clear form of the passcode, generate a back end encrypted passcode, and communicate the back end encrypted passcode to an authentication network.

[0010] In another aspect, the disclosure includes a secure passcode authentication system. The system can include an Access Control Server (ACS) configured to receive a request for Personal Identification Number (PIN) authentication of a Primary Account Number (PAN), and configured to generate a request for a PIN corresponding to the PAN. The request for the PIN can include hidden fields including a unique transaction identifier and a hash value. The system can also include a front end Hardware Security Module (HSM) coupled to the ACS, and configured to generate the hash value based in part on the unique transaction identifier, and further configured to receive an encrypted PIN, decrypt the PIN to recover a clear form of the PIN, and generate a local encrypted PIN using a local encryption key, and a back end HSM configured to receive the local encrypted PIN from the front end HSM and further configured to recover a clear form of the PIN from the local encrypted PIN, generate an Acquirer Working Key (AWK) encrypted PIN, and communicate the AWK encrypted PIN to an authentication network.

[0011] In still another aspect, the disclosure includes a secure passcode authentication system. The system can include an Access Control Server (ACS) configured to receive a request for Personal Identification Number (PIN) authentication of a Primary Account Number (PAN), and configured to generate a request for a PIN corresponding to the PAN, the request for the PIN including an instruction to provide the PIN to a destination address, and a front end Hardware Security Module (HSM) having the destination address and coupled to the ACS, and configured to receive an encrypted PIN, decrypt the PIN to recover a clear form of the PIN, and generate an

Acquirer Working Key (AWK) encrypted PIN using an AWK encryption key, and configured to communicate the AWK encrypted PIN to an authentication network.

[0012] In yet another aspect, the disclosure includes a method of secure passcode authentication. The method can include requesting a Personal Identification Number (PIN) corresponding to a Primary Account Number (PAN), receiving the PIN in response to the request, generating a PINBLOCK based in part on the PIN, encrypting the PINBLOCK using a local key in a front end Hardware Security Module (HSM) to generate a local key encrypted PINBLOCK, decrypting the local key encrypted PINBLOCK with a back end HSM, generating a back end encrypted PIN with the back end HSM, communicating the back end encrypted PIN to an authentication network, and receiving an authentication response from the authentication network.

[0013] In yet another aspect, the disclosure can include a method of secure passcode authentication. The method can include receiving an encrypted Personal Identification Number (PIN) corresponding to a Primary Account Number (PAN), decrypting the encrypted PIN in a front end Hardware Security Module (HSM) to generate a clear form of the PIN, generating a PINBLOCK based in part on the clear form of the PIN, generating in a back end HSM a back end encrypted PIN based in part on the PINBLOCK, communicating the back end encrypted PIN to an authentication network, and receiving an authentication response from the authentication network.

[0014] In yet another aspect, the disclosure can include a method of secure passcode authentication. The method can include generating encryption data, querying a cardholder for a Personal Identification Number (PIN) corresponding to a Primary Account Number (PAN), receiving an encrypted PIN and at least a portion of the encryption data in response to the query, generating a clear form of the PIN based in part on the encrypted PIN, generating a PINBLOCK based in part on the clear form of the PIN, encrypting the PINBLOCK in a front end Hardware Security Module (HSM) using triple DES encryption to generate an encrypted PIN (EPIN), decrypting the EPIN in a back end HSM to recover the clear form of the PIN, encrypting the clear form of the PIN in the back end HSM using an Acquirer Working Key (AWK) to generate an AWK encrypted PIN, communicating the AWK encrypted PIN to an authentication network, and receiving an authentication response.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] The features, objects, and advantages of embodiments of the disclosure will become more apparent from the detailed description set forth below when taken in conjunction with the drawings, in which like elements bear like reference numerals.

[0016] Figure 1 is a functional block diagram of an embodiment of a secure authentication system.

[0017] Figure 2 is a functional block diagram of an embodiment of a communication device that can be used in the secure authentication system.

[0018] Figure 3 is a flowchart of an embodiment of a secure authentication process.

[0019] Figure 4 is a flowchart of an embodiment of a passcode authentication process.

[0020] Figure 5 is a flowchart of an embodiment of a passcode authentication process.

[0021] Figure 6 is an embodiment of a passcode entry form used in one embodiment of a passcode authentication process.

[0022] Figure 7 is an example of encryption data fields used in an embodiment of a secure authentication process.

[0023] Figure 8 is an example of return encryption fields used in an embodiment of a secure authentication process.

[0024] Figure 9 is a flowchart of an embodiment of a secure authentication process.

[0025] Figure 10 is a flowchart of an embodiment of a method of encoding the RDIR_URL to include an EPIN.

DETAILED DESCRIPTION OF THE INVENTION

[0026] The invention in the form of one or more exemplary embodiments is described below. A method and system for secure authentication of payment cards in an electronic commerce environment is disclosed. A merchant can reduce consumer fraud in electronic commerce transactions by authenticating payment card numbers presented by a customer. The merchant

can authenticate the payment card numbers using an issuer authentication network. A customer that is a party to an electronic commerce transaction can ensure that a passcode, such as a PIN, password, or some other authentication code submitted to an electronic commerce site will remain secure.

[0027] A merchant can establish an electronic commerce site on a public network, such as the Internet. A customer can access an electronic commerce web site and shop for goods or services. After selecting desired goods and services, the customer can proceed to checkout in a process that can be similar to the checking out process used in a physical merchant location. The customer can elect to pay for the desired goods and services using a credit card, debit card, stored value card, and the like, or some other type of payment card. The customer can submit, for example, the credit card or debit card number to the merchant.

[0028] The merchant can receive the payment card number and submit the number to an issuing authority to have it authenticated. Additionally, especially in the case of debit cards, the customer may need to provide a passcode, such as a password, an authentication code, or PIN to successfully authenticate the card number.

[0029] As noted above, the communication system used to connect the customer to the merchant can be a public network such as the Internet. However, an issuer authentication network may be a private network. Additionally, the issuer authentication network may require the customer payment card number or passcode information to be supplied using a protocol different from the communication protocol used in the connection between the customer and the merchant. The issuer authentication network may utilize an encryption key to encrypt information communicated across the network. The issuer authentication network can maintain the security of the network by not exposing the encryption key outside of the network.

[0030] However, because the merchant is unable to provide the encryption key to a customer, the payment card number and any associated passcode may need to be recovered prior to encrypting with the encryption key used by the issuer authentication network.

[0031] The customer typically prefers the passcode not appear in clear form, that is in unencrypted form, except when decrypted by the issuer. The disclosed system and method allows a payment card having a passcode received in a front end system using a front end

communication protocol to be authenticated in an issuer authentication network using an encryption key and communication protocol that is not exposed to the customer.

[0032] The disclosed system and method allow a payment card number and associated passcode to be received in a front end system and securely handed to a secure back end system while substantially minimizing or eliminating storage of the customer information in clear form. The disclosed system and method also minimize exposure of the back end security measures outside of the back end system.

[0033] Figure 1 is a functional block diagram of an embodiment of the secure authentication system 100. The system 100 includes a cardholder device 110 in communication with a merchant device via a network 102. In one embodiment, the cardholder device 110 includes a browser 112 running on a computer. The merchant device can be, for example, a merchant server 120 hosting an electronic commerce application. The merchant server 120 can include a merchant module 122 configured to perform portions of the electronic commerce functions, as will be described in further detail below. In an embodiment, the cardholder device 110 can connect over a network 102, such as the Internet, to the merchant server 120 and engage in an electronic commerce transaction.

[0034] The system 100 can also include a directory server 130 coupled to the network 102. The directory server 130 can be configured to determine whether a particular payment card number is within a range of numbers participating in the authentication system 100. An Access Control Server (ACS) 140 can also be coupled to the network 102. The ACS 140 can be coupled to a first Hardware Security Module (HSM1) 142 and a second Hardware Security Module (HSM2) 144. The hardware security modules 142 and 144 may also be referred to as host security modules because the hardware security modules may be implemented in hardware, software, or a combination of hardware and software. The hardware security modules 142 and 144 can be used to encrypt the information provided by the ACS 140 over the network 102. The ACS 140 can also be coupled to a database 146 that is used to store information relating to the participating payment card numbers, such as their associated passcodes.

[0035] A network payment gateway, which may be an Internet Payment Gateway Server (IPGS) 150 can also be coupled to the network 102. The IPGS 150 can manage authentication and acceptance of payment messages sent over the network 102. The IPGS 150 can provide an

interface between the network 102 and an issuing or acquiring institution. The issuer module 160 can be coupled to the IPGS 150. In one embodiment, the IPGS 150 is coupled to the issuer module 160 via a network 155. The network 155 may be a private network or network having limited access. The issuer module 160 can be configured to provide a payment processing system that can exist for multiple types of transactions, and may not be limited to electronic commerce transactions.

[0036] In one example, a customer conducts a transaction in accordance with one embodiment of the secure authentication system 100 of Figure 1. The customer, also referred to as the cardholder, can use the cardholder device 110 to visit a merchant website hosted on the merchant server 120. The cardholder device 110 can communicate with the merchant server 120 using a browser 112 coupled to the cardholder device 110. The customer can select desired goods or services from the merchant website and can proceed to a merchant's checkout page. At the checkout page, the cardholder device 110 and merchant server 120 may initiate a SSL session to encrypt or otherwise secure the personal information communicated over the network 102 by the cardholder device 110. The merchant server 120 can receive the relevant information transmitted by the cardholder device 110.

[0037] For example, the customer can provide a credit card number, debit card number or some other type of payment card number to one or more fields in the cardholder browser 112. The payment card number may be referred to generally as a Primary Account Number (PAN). The customer can then submit the information, typically including the PAN, customer name, and shipping address, to the merchant server 120. The cardholder device 110 can encrypt the information according to the SSL protocol before transmitting the information over the network 102 to the merchant server 120.

[0038] The merchant server 120 can receive the SSL encrypted messages and can recover the clear text, or non-encrypted information, from the messages. The merchant server 120 can use the merchant module 122, alternatively referred to as a merchant plug-in, to query the directory server 130 to verify the customer's eligibility for participation in the secure authentication system 100. Customers having card numbers that are not eligible for the secure authentication system 100 may use an alternative system or may rely on the security provided by the SSL protocol for transaction security. The directory server 130 may query an internal database, an associated

database (not shown) or may query the ACS database 146 to determine eligibility of the customer's card. If the directory server 130 determines the PAN is in a participating card range, the directory server 130 can query the appropriate issuer ACS 140 to validate customer participation. The directory server 130 receives an indication from the ACS 140 and can send a response back to the merchant server plug-in 122.

[0039] The merchant server module 122 can send an authentication request to the ACS 140. In one embodiment, the merchant server module 122 sends the request via the cardholder device 110 and browser 112. The ACS 140 can determine, based in part on the PAN, if a PIN, a password, or some other type of passcode is to be used to authenticate the transaction.

[0040] If the ACS 140 determines that password-based authentication is used, the ACS 140 can query the customer for the password. The ACS 140 can transmit a query to the browser 112 in the cardholder device 110. The customer can enter the password and transmit it to the ACS 140 using the browser 112 in the cardholder device 110. The ACS 140 can authenticate the customer and PAN locally, for example, using information stored in the database 146.

[0041] If the ACS 140 determines that PIN-based authentication is used, the ACS 140 can query the customer for the PIN. The ACS 140 can transmit a query to the browser 112 in the cardholder device 110. The customer can enter the PIN and transmit it to the ACS 140 using the browser 112 in the cardholder device 110. The ACS 140 can convert the received PIN into an Acquirer Working Key (AWK) encrypted PINBLOCK using, for example, the first and second HSMs 142 and 144. The ACS 140 can send the AWK-encrypted PINBLOCK through an IPGS 150 and a network 155 to the issuer module 160 for authentication.

[0042] As noted above, the ACS 140 can perform local authentication of the password. For example, an SHA-1 hash of the password can be stored local to the ACS 140 such as in the database 146. The ACS 140 typically does not locally perform PIN-based authentication. Instead the system 100 may implement remote authentication where the authenticator is not local at the ACS 140, but may be a member bank or card issuer having a issuer module 160 that is remote from the ACS 140. This authentication may occur via the IPGS 150.

[0043] The ACS 140, whether performing local authentication or remote authentication, can return an authentication response to the merchant server module 122 via the customer browser

112 and cardholder device 110. The ACS 140 may also pass a record of the authentication to an authentication history server (not shown). The merchant server plug-in 122 may validate the authentication response. A payment authorization may take place if authentication was successful.

[0044] Although the various blocks in the system of Figure 1 and in subsequent figures can be implemented as hardware modules, one or more of the modules may be implemented as software stored in one or more storage devices and executed by one or more processors. The various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both.

[0045] To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. The functions may be capable of implementation in varying ways for each particular application, but particular implementations should not be interpreted as causing a departure from the scope of the disclosure.

[0046] For example, the browser 112 may be a software application that runs on the cardholder device 110. The cardholder device 110 may be a computer, such as a personal computer, notebook computer, personal digital assistant, or some other device for computing. The cardholder device 110 may also be a point of sale device, a kiosk, a telephone such as a wireless telephone, or some other communication device.

[0047] Figure 2 is a functional block diagram of a communication device 200. Similar communication devices 200 can be used, for example as the cardholder device 110 or one of the various servers in the authentication system 100 of Figure 1.

[0048] The communication device 200 can include a display 210, I/O devices 250 including a keyboard 252 and an input device 254, a processor 220, memory 224, an I/O controller 240, a hard drive 262, one or more removable storage drives 264, which can include a floppy drive, an optical storage 266, some other storage devices 268, a communication device 230 such as a

modem, and a network interface card (NIC) 234. The various elements can be coupled using one or more computer busses 202 within the communication device 200. The one or more storage devices 268 can include, but are not limited to, ROM, RAM, non-volatile RAM, flash memory, magnetic storage, optical storage, tape storage, hard disk storage, and the like, or some other form of processor readable media.

[0049] Figure 3 is a flowchart of an embodiment of a method 300 of secure authentication that may be implemented in the system 100 of Figure 1. The method 300 begins at block 302 where a customer, for example using an Internet browser running on a personal computer, engages in an electronic commerce transaction with a merchant having an Internet accessible site. The customer can select desired goods or services and can proceed to a checkout page on the merchant web site.

[0050] Upon proceeding to the checkout page, the system proceeds to block 310 where the merchant server initiates an SSL connection with the customer computer, for example, by submitting payment pages which are served on a host requiring SSL. The merchant server may require an SSL connection because the checkout page may request that the customer provide confidential personal information, including name, address, type of payment card, and PAN. Because the Internet is a public network, the merchant server and customer computer may set up the SSL connection in order to secure the data exchanged during the checkout process.

[0051] After setting up the SSL connection, the system proceeds to block 320 where the merchant server queries the customer for the customer data and receives the customer data in response to the query. After receiving the customer data, which typically includes a PAN for electronic payment, the system proceeds to block 330 where the merchant server determines whether the PAN received from the customer is configured for secure authentication. The merchant server can query, for example, a database or directory server to determine if the PAN is configured for secure authentication. The directory server may return a response message to the merchant server indicating the configuration of the payment card.

[0052] The system proceeds to block 332 if the customer card is configured for secure authentication. The process steps for a card not participating in secure authentication is not shown. At block 332, the merchant server sends an authentication request to an appropriate ACS. The appropriate ACS may be determined, for example, based in part on the PAN.

[0053] The system then proceeds to block 340 where an issuer ACS may query the cardholder for a passcode. The passcode may be, for example, a password, PIN, and the like, or some other customer security identifier. The system then proceeds to decision block 350 and determines the type of passcode associated with the card. The ACS can determine, for example, whether the particular payment card uses a password or PIN for authentication purposes. The ACS may determine the type of passcode associated with the card based in part on the PAN.

[0054] If the ACS determines a passcode is the type of password, the ACS proceeds to block 352 and performs password authentication. To authenticate a password, the ACS may query the customer for the password and receive the password in response. The ACS can then authenticate the password. In one embodiment, the ACS determines a Secure Hashing Algorithm (SHA-1) hash of the password and compares the value to a previously stored hash value. If the two compare, the password is authenticated. The system proceeds to block 360.

[0055] Returning to decision block 350, if the ACS determines the passcode is a PIN, such as when the customer submits a debit card number for payment, the system proceeds to block 354 and the system performs PIN authentication. Flowcharts of different PIN authentication embodiments are provided in Figures 4 and 5. After authenticating the PIN, the system proceeds to block 360.

[0056] After authenticating the passcode, the ACS, in block 360, returns an authentication message to the requesting merchant server. The system then proceeds to block 370 where the merchant server receives the authentication message from the ACS and validates the message. The ACS may validate the message in order to minimize the likelihood of false authentication messages. The ACS may validate the message for example, by verifying certain parameters or fields that may be included in the authentication message. For example, a digital signature created by the ACS may be validated by the merchant server. In addition, a unique identification number generated for each authentication may be generated by the ACS in the authentication request and the ACS may verify that the unique identification number is included in a field of the authentication message. The desired field may be encrypted to minimize the possibility of a false field value.

[0057] After validating the authentication message, the merchant server proceeds to block 380 if the passcode authenticates the user PAN. In block 380 the merchant server initiates payment

authorization. If the passcode does not authenticate the PAN, the system may proceed to an error process (not shown).

[0058] Figure 4 is a flowchart of an embodiment of a PIN authentication process 354, such as the PIN authentication process of Figure 3. The process 354 begins at block 402 where the ACS queries the customer, via the browser and cardholder device, for the PIN corresponding to the previously submitted PAN. The ACS then proceeds to block 410 to wait for and receive the PIN from the cardholder device.

[0059] The ACS and cardholder device initiate a new SSL connection that is independent of the SSL connection between the cardholder device and the merchant server, the customer PIN supplied by the customer is not exposed to the merchant server, and is not recoverable by the merchant server using the SSL keys currently used by the merchant server to communicate with the cardholder device.

[0060] Once the ACS receives the PIN from the cardholder device, the ACS proceeds to block 420 to generate a PINBLOCK. The ACS can generate the PINBLOCK, for example, using a combination of PAN and PIN values, solely PIN values, or a combination of PIN values and other information. The PINBLOCK can be, for example, a predetermined format for communicating PIN information. In one embodiment the PINBLOCK is an ISO 9564 Format-0 PINBLOCK, also referred to as an ANSI X9.8 Format-0 PINBLOCK or an RG7100 Format 05 PINBLOCK. The ACS may need to recover the SSL encrypted PIN in order to generate the PINBLOCK. Thus, there is a possibility that the PIN is available in clear, non-encrypted, form in the ACS memory.

[0061] The ACS then proceeds to block 430 and encrypts the PINBLOCK using a predetermined encryption algorithm. For example, the encryption algorithm may be stored in a first hardware security module (HSM1). The first HSM can store a local Zone Protection Key (ZPK). The HSM can be configured to encrypt the PINBLOCK using the local ZPK. The encryption algorithm may be, for example, a symmetric or asymmetric key based encryption algorithm. In one embodiment, the encryption algorithm is a DES encryption algorithm.

[0062] The first hardware security module (HSM1) can be implemented in hardware or in software. If the HSM1 is implemented in software, the HSM1 can be implemented within the

ACS such that the SSL decryption and PINBLOCK generation is conducted as part of the HSM1. Alternatively, the HSM1 may be implemented in software in a device external to the ACS. Thus, the clear form of the PIN may only be available in memory within the HSM1. Because the HSM1 can be configured to handle data securely, such as in accordance with Federal Information Processing Standards (FIPS) security standards, the clear form of the PIN in the HSM1 may only exist in a temporary register and therefore poses a minimal security hazard.

[0063] The system proceeds to block 440 and the PINBLOCK that is encrypted by the HSM1 is re-encrypted using the second Hardware Security Module (HSM2). The HSM2 can be configured to decrypt the HSM1 encrypted PINBLOCK prior to re-encryption. In an embodiment, the software encrypted PINBLOCK is provided to HSM2 which is implemented in hardware. In another embodiment, HSM2 can be implemented as a Thales RG7100 hardware security module. The HSM2 can be configured to decrypt the HSM1 encrypted PINBLOCK and encrypt the PINBLOCK using an Acquirer Working Key (AWK) to generate an AWK encrypted PINBLOCK.

[0064] The system proceeds to block 450 where the ACS communicates the AWK encrypted PINBLOCK to an issuer for authentication. In one embodiment, the ACS communicates the AWK encrypted PINBLOCK to an Internet Payment Gateway Server (IPGS). The IPGS can be coupled to an issuer server and may communicate the AWK encrypted PINBLOCK to the issuer for authentication.

[0065] The system proceeds to block 460 where the ACS waits for and receives an authentication response from the issuer. In the embodiment discussed above, the issuer may send an authentication message through an IPGS, and the IPGS may transmit the message to the ACS. The use of an IPGS may be particularly advantageous where a first network connects the ACS to the IPGS and a second network connects the IPGS to the issuer.

[0066] Figure 5 is a flowchart of another embodiment of a PIN authentication process 354, such as the PIN authentication process of Figure 3. The process described in Figure 5 allows the PIN to be transmitted from the cardholder device directly to an HSM and does not expose the PIN in clear form outside of an HSM. The ACS receives encrypted versions of the PIN and may not have the ability to recover the clear form of the PIN.

[0067] The process 354 begins at block 502 where the ACS in combination with the HSMs generate data that will be used in subsequent encryption steps. The data can include, for example, a unique transaction ID that is generated for the authentication transaction, a destination address, a redirect address that can be used to redirect a communication to a desired destination address, a redirect type such as a type that can be used with http type redirect instructions, and a message authentication code that may be generated by one or more HSMs that will be used in the encryption process.

[0068] Once the encryption data is generated, the system proceeds to block 504 where the ACS sends a query to the cardholder device asking the cardholder for the PIN. The query can include the previously generated encryption data. The encryption data may be hidden, such that the customer does not view the information.

[0069] The system then proceeds to block 510 to wait for the customer to submit the PIN. The customer can input the PIN and submit the PIN to the system. The PIN and previously generated encryption data are then sent directly to the first, or front end HSM (HSM1) whose address can be included in the encryption data as a destination address. Thus, although the ACS can generate and send the PIN request to the cardholder device, the cardholder device can be instructed to send the PIN directly to an HSM. Additionally, the cardholder device and HSM1 can initiate a new SSL connection. A new SSL connection between the cardholder device and the HSM1 allows the communication between the two devices to be secure, and can be secured even with respect to the ACS that initiated the communication. Thus, in block 510, the HSM1 receives the PIN and at least a portion of the previously generated encryption data.

[0070] The system then proceeds to block 530 where the HSM1 generates a PINBLOCK. In one embodiment, the HSM1 is a hardware HSM. The HSM1 can be configured to generate an ISO Format-1 PINBLOCK using the PIN. In another embodiment, the HSM1 can generate an ISO Format-1 PINBLOCK using the PIN and PAN, where the PAN can be supplied from the customer or the ACS. The HSM1 recovers the clear form of the PIN from the SSL encrypted message, but the clear form is maintained in accordance with the security of the HSM1. For example, the HSM1 may provide security in accordance with FIPS 140-1, FIPS 140-2, or some other security standard. In one embodiment, the HSM1 is configured to provide a level of

security in accordance with FIPS 140-2, level 3. In another embodiment, the HSM1 is configured to provide a level of security in accordance with FIPS 140-2, level 2.

[0071] After the PINBLOCK is generated by the HSM1, the system proceeds to block 530 and the HSM1 generates an encrypted PIN (EPIN). In one embodiment, the HSM1 generates an encrypted PIN using triple DES encryption using a local Zone Protection Key (ZPK).

[0072] After the HSM1 generates the EPIN, the system proceeds to block 540 where the HSM1 communicates the EPIN back to the ACS. The HSM1 can, for example, be supplied the destination address of the ACS using the encryption data supplied with the PIN. In one embodiment, the ACS address is included as the redirect address within the encryption data. In another embodiment, the HSM1 communicates the EPIN information and at least a portion of the encryption data to the ACS as arguments in a redirection URL. For example http status codes 302 or 303 may be used to trigger this scenario.

[0073] The system proceeds to block 550 where the ACS extracts the EPIN and desired encryption data received from the HSM1. As described above, the ACS can extract the EPIN and encryption data such as the unique transaction ID from arguments in the URL. The ACS may validate the received information, for example, by comparing the received unique transaction ID against the previously generated value of the ID.

[0074] The system proceeds to block 560. In block 560, the ACS provides the information to the second or back end HSM (HSM2). The HSM2 reformats and re-encrypts the EPIN. The HSM2 can be configured to decrypt the EPIN to recover the clear form of the PIN. The HSM2 can then re-encrypt the PIN using the Acquirer Working Key to produce an ISO 9564 Format-0 AWK encrypted PINBLOCK. In one embodiment, the HSM1 and HSM2 are separate hardware security modules. In another embodiment, the HSM1 and the HSM2 are implemented in a single hardware security module. In another embodiment, the front end HSM can be an nCipher hardware security module and the back end HSM can be a Thales RG7100 hardware security module. The HSM1 and HSM2 can be co-located or may be remotely located from each other.

[0075] The system proceeds to block 570 where the ACS receives the AWK encrypted PINBLOCK from the HSM2 and communicates the AWK encrypted PINBLOCK to an issuer for authentication. The ACS may send, for example, the AWK encrypted PINBLOCK to an

IPGS. The IPGS may communicate the AWK encrypted PINBLOCK to the issuer for authentication.

[0076] The system then proceeds to block 580 where the ACS waits for the authentication response and eventually receives the authentication response from the issuer. An IPGS may communicate the authentication response to the ACS based on an authentication message received from the issuer.

[0077] Figure 6 is an embodiment of a passcode entry form 600 used in one embodiment of a passcode authentication system, such as the system of Figure 1 implementing the process embodiment of Figure 5. The form is configured as an HTML form that can be sent by the ACS to the cardholder device. The form includes hidden fields that are used to store the previously generated encryption data. The form includes, for example, a field that configures an SSL connection with an HSM1 610. The form may also include a field used to communicate the unique transaction ID 620 or unique authentication session ID. The form may also include fields that identify the redirect address 630 and the redirect type 640. The form can also include a message authentication code 650 that identifies the HSM1 and the authentication session. The message authentication code 650 can be used to ensure that the HSM1 is used by the ACS.

[0078] Figure 7 is an example of a table 700 of encryption data fields used in an embodiment of a secure authentication system. The encryption data can include, for example, a TXNUUID field 710 that can be a unique transaction ID. The encryption data can also include a RDIR_URL field 720 that represents a base URL that is used to redirect communication from the HSM1 back to the ACS. The encryption data can also include a RDIR_TYPE field 730 that can indicate the http redirect type the HSM1 should return. For example, the redirect type can identify 302 or 303 redirect types. The encryption data can also include a GEP_MAC field 740 that represents a message authentication code that can be used to ensure that the HSM1 is used by the ACS. The GEP_MAC field value can be generated, for example, by the HSM1. The HSM1 can be configured to generate the field value by encrypting, for example, the TXNUUID, RDIR_URL, and RDIR_TYPE field values using the local zone protection key.

[0079] Figure 8 is an example of a table 800 of return encryption fields used in an embodiment of a secure authentication system. In this embodiment, the return encryption fields includes an EPIN1 encrypted PIN field. The EPIN1 field can be, for example, an ISO 9564 Format-1

PINBLOCK encrypted using a key that is shared by the HSM1 and HSM2. In an embodiment, the value can be base 64 encoded prior to being used as a field in the redirection URL.

[0080] Figure 9 is a detailed flowchart of an embodiment of a secure authentication process 900 that can be implemented, for example, in the system embodiment of Figure 1. The process 900 begins at block 902 when the ACS receives a payment authentication request. The ACS proceeds to block 904 and retrieves a Primary Account Number (PAN) from, for example, an original verify enrollment request previously received from the merchant server.

[0081] The system then proceeds to block 906 where the ACS retrieves issuer information from a database. The issuer information may be based in part on the PAN and may identify whether the payment card is a credit card or a debit card. The system proceeds to decision block 910 where the ACS determines the type of passcode required to authenticate the payment card. For example, the ACS may determine whether a password or PIN is used to authenticate the card.

[0082] If a password is used, the system may authenticate the card locally using, for example, an SHA-1 hash of the password (not shown). The system proceeds to block 920 if a PIN is used to authenticate the payment card. In block 920, the ACS generates the encryption data that can accompany the PIN request form. The ACS can generate the TXNUUID, RDIR_URL, and RDIR_TYPE values and can initiate generation of a hashed message authentication code (HMAC) value. The RDIR_URL can be the URL of the ACS. The value of the RDIR_URL may be modified in a later process step.

[0083] The ACS can transmit a message to the HSM1 to initiate generation of the HMAC value. The HSM1 can be referred to as a front end HSM because the HSM interfaces with the ACS, and may interface with the cardholder device.

[0084] The HMAC value can be the value used in the GEP_MAC field of the encryption data, as shown in the table of Figure 7. The HSM1 can generate the HMAC value, for example using a first encryption key (k1) stored within the HSM1. The HSM1 can generate the HMAC value, for example, by generating a hash of the TXNUUID, RDIR_URL, and RDIR_TYPE values using the k1 key. The HMAC value can be used as a form of one-way authentication. The HSM1 generates the HMAC and later verifies a received HMAC to determine if it compares

with the original HMAC value. Thus, the HMAC value does not contain any values that need to be extracted outside of the HSM1. In fact, the HSM1 may regenerate the HMAC value to verify the authenticity of a message received at the HSM1.

[0085] Once the HSM1 generates the HMAC value, it is provided to the ACS. The ACS inserts the HMAC value in the encryption data and generates the PIN request form. The system proceeds to block 924 where the ACS sends the PIN request form to the cardholder device. The form can be configured such that when the customer submits the form the information is directed to the HSM1 and does not need to be routed through the ACS.

[0086] The system proceeds to block 930 where the customer receives the PIN request form at the cardholder device. The customer can enter the PIN and submit the form to the HSM1. The system proceeds to block 932 when the form is submitted by the customer.

[0087] At block 932 the cardholder device establishes an SSL connection with the HSM1 such that the information provided by the customer, via the cardholder device, can be transmitted in an encrypted form, thereby avoiding exposing the PIN and other personal information in clear form over a public network. After establishing the SSL connection with the cardholder device, the HSM1 proceeds to block 940.

[0088] In block 940, the HSM1 retrieves the encryption data from the cardholder message. After the HSM1 obtains the encryption data, the system proceeds to block 942.

[0089] In block 942 the HSM1 verifies the retrieved GEP_MAC data. The HSM1 can, for example, re-generate the HMAC value initially generated in block 922 and compare the value to the HMAC value received from the cardholder device. The HSM1 can use the same values and the same key used to generate the initial HMAC value. There is low probability of message tampering or corruption if the two HMAC values are substantially identical.

[0090] The GEP_MAC value can be used to ensure that HSM1, which typically creates the encrypted PIN, processes transactions which originated from the ACS. The GEP_MAC verification can be used to avoid random users from hitting HSM1 and tying up the resource.

[0091] After verifying the GEP_MAC value, the system proceeds to block 950. At block 950 the HSM1 retrieves the PIN value from the received data. The system proceeds to block 952 and

generates a PINBLOCK. The HSM1 can generate a PINBLOCK using a predetermined PINBLOCK format such as a Format-1 PINBLOCK.

[0092] The system proceeds to block 954 and generates the encrypted PIN value (EPIN) using a second key value (k2) stored within the HSM1. The second key value (k2) may also be referred to as the back end key. The HSM1 can generate the EPIN value using any number of encryption algorithms. In one embodiment, the HSM1 encrypts the PINBLOCK using triple DES. Alternatively, the HSM1 may use Advanced Encryption Standard (AES), Data Encryption Standard (DES), RSA, Skipjack, Blowfish, SEAL, RC-4, and the like, or some other encryption algorithm.

[0093] The second key value (k2) can be a key value that is shared with the HSM2. The second key value (k2) may be a key value that is used to communicate information between the two HSMs and may not be used for communicating with other modules. Thus, although the HSM1 and HSM2 may be located remote from one another, the second key (k2) may be referred to as a local key.

[0094] After the HSM1 generates the encrypted PIN value, the system proceeds to block 956 and the HSM1 encodes the redirect URL. In one embodiment, the HSM1 may also optionally generate a new HMAC value using the k1, TXNUUID, and EPIN values. In one embodiment, the RDIR_URL value can be generated by the ACS using, for example the process shown in Figure 10.

[0095] The new HMAC value, which may be referred to as a REP_MAC value, can be used to prevent replay attacks and provide idempotency. That is, the REP_MAC helps prevent the authentication data such as the encrypted PIN from being re-used by the browser such as if a user scrolls back through a browser history. The REP_MAC value can also be used to ensure that the encrypted PIN (EPIN) was generated by HSM1.

[0096] After generating the RDIR_URL value, the system proceeds to block 960 where the HSM1 generates a http 302/303 redirect to the redirect URL. The HSM1 can include the TXNUUID and encrypted PIN values as arguments in the URL. The HSM1 communicates the RDIR_URL to the cardholder device.

[0097] The system proceeds to block 962 where the cardholder device receives the RDIR_URL and handles the message redirection. The cardholder redirects the message to the redirect URL, which is typically the address of the ACS.

[0098] The system proceeds to block 970 where the ACS receives the redirected message from the cardholder device. The ACS can extract the TXNUUID and EPIN values from the message. For example, the ACS can extract the TXNUUID and EPIN values from the URL arguments.

[0099] In the system embodiment in which the HSM1 generates an HMAC value and includes the value in the return encrypted data PIN message, the ACS may verify the HMAC value. To verify the HMAC value, the ACS may extract the HMAC value and communicate the value to the HSM1. The HSM1 can receive the HMAC from the ACS and can regenerate the HMAC value with the k1, TXNUUID, and EPIN values. The HSM1 can then compare the received HMAC value to the re-generated HMAC value and can provide the ACS a verification or validation message if the two are substantially the same.

[0100] The system then proceeds to block 972 where the ACS reconciles the TXNUUID with the previous values generated by the ACS. The system proceeds to block 974 and submits the PAN, EPIN, and any other values used to generate an AWK encrypted PINBLOCK to the HSM2.

[0101] The system proceeds to block 980 where the HSM2 receives the information provided by the ACS. The HSM2 can recover the clear form of the PIN by decrypting the EPIN using the shared second key value (k2). The HSM2 may reformat the PIN in another PINBLOCK format. The HSM2 may also be referred to as the back end HSM because the HSM2 is used to communicate with the issuer and does not typically communicate with the cardholder device or merchant server.

[0102] The HSM2 can generate an AWK encrypted PINBLOCK using the AWK that may be stored within the HSM2. The AWK may be a third key (k3) used by the system and may be used by the issuer network to secure communications. The HSM2 may use any type of encryption algorithm, such as one or more of those that may be implemented in the HSM1. The HSM2 communicates the AWK encrypted PINBLOCK to the ACS.

[0103] The system proceeds to block 990. In block 990, the ACS receives the AWK encrypted PINBLOCK and submits the PINBLOCK to the issuer for authentication. In one embodiment, the ACS communicates the AWK encrypted PINBLOCK to an IPGS and the IPGS communicates the AWK encrypted PIN to an issuer for authentication.

[0104] Figure 10 is a flowchart of an embodiment of a method 1000 of encoding the RDIR_URL. The method 1000 can be implemented, for example, in the HSM1 of the system of Figure 1.

[0105] The method 1000 begins at block 1010 where the HSM1 determines the EPIN value. As noted in the earlier flowchart discussions, the HSM1 can generate an encrypted PIN value that is communicated to the HSM2. The HSM1 may encode the RDIR_URL as part of the EPIN generation process, or may encode the RDIR_URL one or more process steps after generating the EPIN. The HSM1 may, for example, retrieve the EPIN from memory within the HSM1.

[0106] The HSM1 then proceeds to block 1020 and base 64 encodes the EPIN value. After base 64 encoding the EPIN value the HSM1 proceeds to block 1030 and appends the base 64 encoded value to a string having one or more characters, such as "epin1=".

[0107] After appending the base 64 encoded EPIN to the string, the HSM1 proceeds to block 1040 and appends the string to the RDIR_URL previously generated by the ACS. The string value can be an argument of the RDIR_URL. The HSM1 then proceeds to block 1050 and generates an http redirect using the modified RDIR_URL as the redirect address.

[0108] The disclosed system and methods can be used to securely authenticate a passcode, such as a payment card PIN. The system can authenticate a payment card PIN received over a public network without exposing the clear form of the PIN to a publicly accessible device. The PIN may be received from the customer using an encrypted format over a public network. In one embodiment, the system can receive a customer PIN using a SSL connection to communicate the PIN over the Internet. The secure authentication system can reformat the PIN by stripping the front end encryption and applying a back end encryption without exposing the clear form of the PIN.

[0109] Although specific embodiments have been described in the above block diagrams and flowcharts, the system may be modified without departing from the scope of the disclosure. In

one such modified embodiment, a single HSM is used in place of two distinct HSMs. Alternatively, the functions performed by the HSM1 and HSM2 can be implemented within a single HSM.

[0110] If a single HSM is used, the intermediate step of encoding the recovered clear form of the PIN using a local ZPK may be eliminated. Rather, a single HSM can be configured to recover the clear form of the PIN, such as by decrypting the SSL encrypted PIN provided by the ACS or the cardholder device. The HSM can then generate the AWK encoded PINBLOCK without generating the EPIN using the intermediate local ZPK and intermediate encryption algorithm.

[0111] The above description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the disclosure. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without departing from the scope of the disclosure. Thus, the disclosure is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.